

*Citation for published version:*

Feldman, J, Hicks, B & Mullineux, G 2014, 'Mechanism synthesis: a comparison of adaptive and traditional approaches', *International Journal of Information Technology and Management*, vol. 13, no. 2/3, pp. 216-241.  
<https://doi.org/10.1504/IJITM.2014.060308>

*DOI:*

[10.1504/IJITM.2014.060308](https://doi.org/10.1504/IJITM.2014.060308)

*Publication date:*

2014

*Document Version*

Early version, also known as pre-print

[Link to publication](#)

© 2014 Inderscience Publishing

**University of Bath**

**Alternative formats**

If you require this document in an alternative format, please contact:  
[openaccess@bath.ac.uk](mailto:openaccess@bath.ac.uk)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# A comparison of optimization methods for mechanism synthesis

Jonathan A. Feldman

Glen Mullineux \*

Ben J. Hicks

Innovative Design and Manufacturing Research Centre,  
Department of Mechanical Engineering,  
University of Bath,  
Bath BA2 7AY,  
United Kingdom.

## Abstract

*The synthesis problem is a critical stage in the design of mechanisms. It can be formulated as an optimization problem and ways to solve this have been the subject of much recent research. A number of methods based on the ideas of adaptive computing have emerged. This interest in this paper is a comparison of these approaches with the conventional direct search and gradient methods. This comparison is made by considering three case study examples, based on four bar mechanisms, and comparing the results obtained by conventional means with those published for the newer methods. It is found that the conventional methods perform at least as well as the others and arguably involve a more simple and intuitive problem formulation.*

---

\*Corresponding author: Address: Innovative Design and Manufacturing Research Centre, Department of Mechanical Engineering, University of Bath, Bath BA2 7AY, United Kingdom; e-mail: [g.mullineux@bath.ac.uk](mailto:g.mullineux@bath.ac.uk)

# 1 Introduction

The problem of synthesising a mechanism to output a prescribed path is an important one in the area of mechanism design [1, 2, 3, 4]. Given an output path, specified normally as a collection of discrete precision points, the task is to find a mechanism whose output traverses these points exactly or, at least, as closely as possible. There may additionally be restrictions in terms of specifying the required value of the drive angle for each of the precision points.

The problem can of course be formulated as a collection of non-linear equations each of which corresponds to one of the precision points. This formulation, in turn, can be recast by forming the sum of the squares of the “errors” associated with each equation. This is treated as a function of the mechanism parameters and a configuration sought which makes the function zero or, at least a (local) minimum.

In this way, the synthesis task becomes an optimization problem. A variety of different numerical techniques can be applied. These include the “traditional” direct search and gradient methods [5, 6, 7, 8], as well as adaptive computing methods such as genetic algorithms [9, 10, 11, 12, 13], and methods based around constraint programming [14].

Given this array of different approaches, two questions arise. The first concerns how the various methods compare and whether there, in some sense, a best method for the path synthesis problem. The received wisdom is that “traditional” search methods depend heavily upon a reasonably good seed mechanism from which to start if they are to home in on a local optimum. In contrast, adaptive methods are able to search over a much wider region of feasible designs and hence are more likely to reach a global optimum. However the resolution of adaptive methods may be less good: that is to say, they are less able to determine the

precise optimum within the local space of the global solution when compared to a traditional method given an appropriate seed. This suggests an approach of using an adaptive method to find the general region in which the global optimum lies and then a more traditional method to find it exactly.

The second question (which is not addressed in any detail in this paper) relates to how mechanism designers work and hence what numerical approaches are best suited to computer aids. Experience over a number of years of working with designers suggests that it is very rare for an entirely new mechanism to be required. Instead, designs migrate from one previously successful design to the next. A radically different design may not be welcome partly because of unfamiliarity with its properties (and potential difficulties) and partly because it may violate implicit constraints on size and avoidance of clashing with other parts of the system. If this is the case, then it suggests that an acceptable approach in practice is to start with an existing successful mechanism and use this as a seed for a local search to satisfy the new design requirements.

One approach to achieving this is to represent a mechanism using constraints (or rules) that govern the relationships and requirements, and then to resolve the constraints and hence determine a configuration that satisfies the requirements. Interest in constraint-based approaches to design has become increasingly apparent in recent years [15, 16, 17, 18, 19, 20]. Again there is a variety of ways in which the constraints can be resolved [21, 22, 23, 24, 25]. For mechanism design, one is interested in an approach which is congruent to the way in which designers wish to work: hence the above question. Additionally the approach needs to be general-purpose as it is convenient to add in additional constraints (not necessarily related to the mechanism itself but to how it interacts with its environment) and the form

of these is not necessarily known a priori [18].

It is also worth noting that the performance of a solution approach is not generally of paramount concern to the user. Since new mechanisms are sought infrequently, whether it takes a few seconds or a couple of minutes to find a good solution does not matter. What is required is robustness of the solution process.

This paper is interested in the first question – comparing the various proposed approaches for the purpose of mechanism synthesis. Advantage is taken of the fact that several recent publications (considering adaptive methods) [10, 11, 13] have appeared which consider the same collection of case study examples. These same examples are here considered using “traditional” approaches in order to create the required comparison.

The next section discusses the formulation of the mechanism synthesis problem in terms of optimization and in section 3 an overview of the various solution methods is provided. Section 4 discusses three different mechanism examples and the results of applying the methods to them. Finally some conclusions are provided.

## 2 Problem formulation

A four bar linkage (4R mechanism) consists of three moving components: the crank, the coupler, and the driven link. The fourth link is the (fixed) base to which the crank and driven links are attached. One way to obtain output from the mechanism is via an off-set point which is part of the coupler. The general arrangement is shown in Figure 1, which also shows the path of the offset point.

The following notation is used here. The (fixed) pivots for the crank and driven links

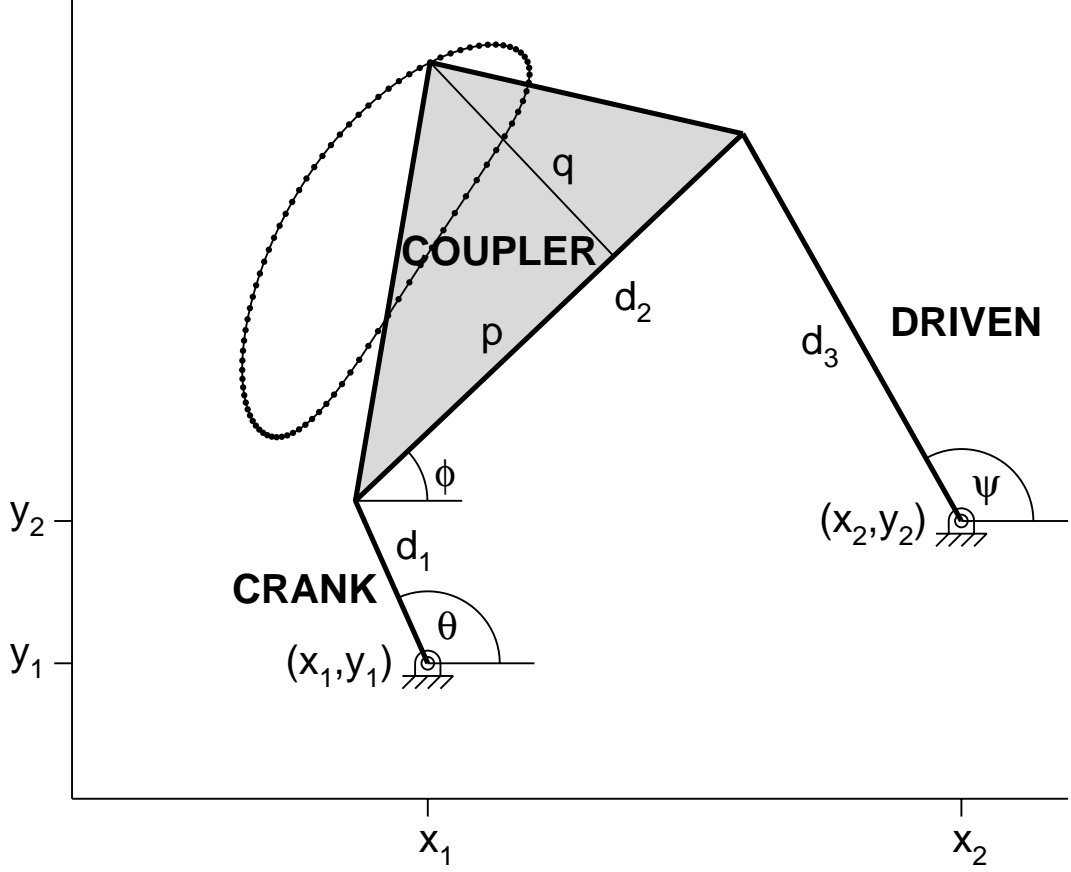


Figure 1: Four bar mechanism and notation

are at positions  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively. The crank, coupler and driven links have lengths  $d_1, d_2, d_3$  respectively. The offset point lies at position  $(p, q)$  relative to a local set of axes for the coupler as in Figure 1. These nine parameters define the mechanism and its offset point.

The angle of the crank, relative to the global  $x$ -axis, is denoted by  $\theta$ . Let  $\phi$  and  $\psi$  be the corresponding angles for the coupler and driven links. Then for a given crank angle, the relations which determine the other two angles are the following.

$$X(\theta) = d_1 \cos \theta + d_2 \cos \phi + d_3 \cos \psi - (x_2 - x_1) = 0$$

$$Y(\theta) = d_1 \sin \theta + d_2 \sin \phi + d_3 \sin \psi - (y_2 - y_1) = 0$$

For any given value of  $\theta$  these equations can be solved in a variety of ways. The easiest is to use the cosine rule in the triangle formed by the ends of the driven link and the free end of the crank. This yields

$$\begin{aligned}\phi &= \gamma \pm \cos^{-1} \left( \frac{+D^2 + d_2^2 - d_3^2}{2d_2D} \right) \\ \psi &= \gamma \pm \cos^{-1} \left( \frac{-D^2 - d_3^2 + d_2^2}{2d_3D} \right)\end{aligned}$$

where

$$\begin{aligned}D^2 &= (x_2 - x_1 - d_1 \cos \theta)^2 + (y_2 - y_1 - d_1 \sin \theta)^2 \\ \gamma &= \tan^{-1} \left( \frac{y_2 - y_1 - d_1 \sin \theta}{x_2 - x_1 - d_1 \cos \theta} \right)\end{aligned}$$

More generally, the ability to assemble as the crank is taken around a full cycle can be assessed as follows. Suppose the crank angle varies over  $m$  values  $\alpha_i$  for  $1 \leq i \leq m$ . Then the expression

$$E_a = \sum_{i=1}^m X(\alpha_i)^2 + Y(\alpha_i)^2 \quad (1)$$

is non-negative and is only zero if the mechanism assembles at each of the  $m$  crank angles. A value of zero only says that the mechanism assembles correctly at each of the  $m$  points considered. It does not guarantee that the mechanism cycles correctly between them, but experience suggest  $E_a$  provides a useful guide provided  $m$  is sufficiently large (say at least 180 with the  $\alpha_i$  equally spaced).

Suppose that a desired output path is specified by means of an ordered collection of  $n$  points,  $\mathbf{R}_i$  for  $1 \leq i \leq n$ . Let  $\mathbf{r}(\theta)$  be the position of the offset point when the crank angle is  $\theta$ , and suppose that angles  $\theta_i$  correspond to the prescribed points on the path. Then the

error in matching the path can be measured by

$$E_p = \sum_{i=1}^n |\mathbf{r}(\theta_i) - \mathbf{R}_i|^2 \quad (2)$$

or by the root mean square (rms) error given as follows.

$$E_r = \sqrt{E_p/n} \quad (3)$$

One approach to finding a mechanism to follow a given path is to consider the combination

$$E = W_p E_p + W_a E_a$$

where  $W_p$  and  $W_a$  are non-negative weighting values. This is a function of the  $n+9$  variables, namely the nine parameters for the mechanism and the  $n$  crank angles corresponding to the path points. A minimum of the combination  $E$  is sought by adjusting the variables. In the examples discussed here, the value of  $W_a$  is taken as zero for simplicity and it is assumed (and later checked) that the mechanism does assemble correctly throughout the cycle.

A variation on the problem is to assume that the crank angles are specified absolutely. In this case,  $E$  is treated as a function of the nine parameters of the mechanism. If the crank angles are specified relative to each other, then there are ten variables to consider in minimising  $E$ .

### 3 Optimization methods

When the path matching problem is expressed as an optimization problem it takes the form

$$\text{minimize } f(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})^2$$



where  $\mathbf{x}$  is a vector of design parameters,  $w_i$  is a positive weighting factor, and each  $f_i(\mathbf{x})$  is a pre-defined function. This form of optimization occurs in other areas of design, particularly constraint-based design, where the functions  $f_i(\mathbf{x})$  relate to the constraints [18, 21, 23, 25, 26, 27, 28]. The nature of the expression means that a least squares problem is being solved.

The interest here is in investigating the performance of several numerical optimization methods that are readily available. These methods need to deal with single-objective, least squares, unconstrained optimization problems.

Such methods can be roughly divided into two classes. The first are the “traditional” methods which are essentially direct search in nature, or which merge into Newton methods when they use numerical approximations for the required derivatives [29, 30, 31]. The second class comprises the more recent methods which are (in some sense) “evolutionary” or “adaptive” [32, 33, 34, 35]. These include methods such as simulated annealing and genetic algorithms and their variations.

The “traditional” methods used are as follows. They are general-purpose methods and were chosen either because of their ease of implementation or because an existing implementation (via a library) was available.

- The method of Hooke and Jeeves (HJ) [29]. This is perhaps the most basic direct search method. The objective function  $f(\mathbf{x})$  is repeatedly evaluated while incrementally changing each of the components of  $\mathbf{x}$  in turn. At each stage the process changes the “current position” to that corresponding to the lowest value of the objective function.

The code used is that implemented by the authors.

- Powell’s direct search method [29]. This is an extension of the method of Hooke and

Jeeves which allows the search directions at each stage to be updated so that one aligns with the expected direction towards an optimum position. The code used is that implemented by the authors.

- The function `fminunc()` provided by MATLAB in its optimization toolbox [36]. This function performs unconstrained optimization using a Quasi-Newton method [30].
- The sequential quadratic programming (SQP) method from the NAG library [37]. This algorithm is for non-linear, constrained, optimization problems and it is based on solving a series of sub-problems which aim to minimize a quadratic model of the objective function subject to a linearization of the constraints.
- The derivative-free bundle method (DFBM) supplied as part of the GANSO (Global And Non-Smooth Optimization) library from University of Ballarat [38]. The DFBM method [39] is intended as an efficient method for finding local minima for problems with non-differentiable objective functions.

The following four evolutionary methods are considered here. They were chosen because they have appeared in recent publications applied to the path matching problem with numerical results being given for the same case study examples.

- Tabu search (tabu) (cf. [11]). This is a direct search method which maintains lists of recently searched positions so as to avoid unwanted repeated searching in these regions. Thus the allowable neighbourhood of the current location is often reduced until the process decides that a previously considered region needs to be revisited.

- The Tabu-Gradient method (tabugrad) [11]. This represents an extension of the basic tabu search method. After each local direct search stage, an additional gradient search is undertaken to encourage better progress towards the (local) minimum.
- The genetic algorithm - fuzzy logic (GA-FL) method [10]. This involves the use of a conventional genetic algorithm. This requires bounding intervals to be imposed on the components of the design vector  $\mathbf{x}$ . The algorithm monitors the progress of each component and uses techniques of fuzzy logic controllers to update these bounds as the search progresses.
- The Smaili-Diab (SD) method [13]. This method is specifically for the synthesis of mechanisms to generate a prescribed path. Its search process is based on an ant-search algorithm [34]. Here a number of search directions are proposed as trails for the “ants” to follow. Depending on how successful these turn out to be in terms of reducing the objective function, some trials are excluded and others are investigated further on a stochastic basis.

## 4 Results

In this section three different case study examples are considered. In each case, a four bar mechanism is sought. The path synthesis problem is solved using the “traditional” methods given in the previous section. The results are compared with results available from the literature for the same examples.

The selection of suitable seed mechanisms as starting points for optimization searches is

of course critical. It is not always clear which were used in the previously published examples. What has been done here is as follows. A catalogue of mechanisms [40] was used in which the required output path could be entered and the closest matches from the catalogue inspected. Such matches can vary considerably in terms of the position of the mechanism relative to the path and the way in which the dyad formed by the coupler and driven links assembles. A choice for a seed was made by selecting a mechanism from the catalogue similar in position and dyad assembly (but of course not exactly the same) to the published results being using for comparison.

For each example, the assessment of resultant mechanisms is in terms of the root mean square (rms) error as given by equation (3). For published mechanisms, the details provided for the geometry were used to simulate the motion and hence obtain the rms error in a consistent way for comparison purposes.

#### 4.1 Problem 1: simple path match (10 points)

The required path [11] is specified by 10 points. These are given in table 1 and are shown in figure 2. The crank angle corresponding to each precision point is also specified. The optimization problem can therefore only vary the nine degrees of freedom of the mechanism.

	1	2	3	4	5	6	7	8	9	10
$x$	1.0	1.5	2.0	2.5	3.0	3.5	4.0	3.5	2.5	1.5
$y$	1.5	2.0	2.0	2.0	2.0	2.0	1.5	1.0	1.0	1.0
$\theta$	0	35	87.5	105	140	175	192.5	262.5	280	315

Table 1: Precision points and prescribed crank angles for problem 1

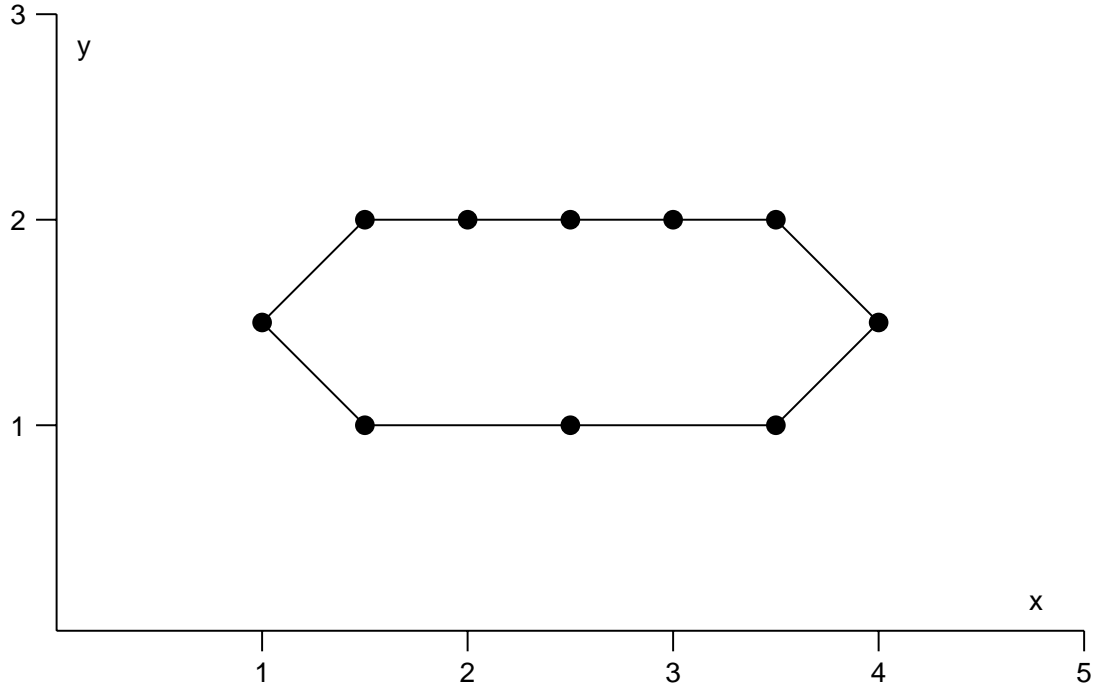


Figure 2: Precision points forming path for problem 1

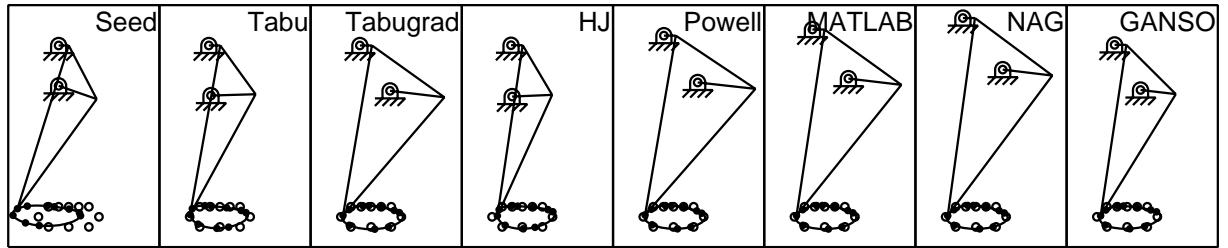


Figure 3: Mechanisms for problem 1

The results are shown in table 2 and they are displayed in figure 3. These show the seed mechanism used for the conventional search processes. Comparison is with the results for the tabu and tabugrad methods [11].

It is seen that all the methods succeed in obtaining good mechanisms in the sense that the values of the rms error  $E_r$  are small. The tabu method performs worst but its variation,

	Seed	Tabu	Tabugrad	HJ	Powell	MATLAB	NAG	GANSO
$x_1$	2.0	1.9500	1.9426	1.9412	2.0231	1.8903	1.8546	1.8798
$y_1$	10.0	10.0000	10.0000	10.0136	10.4883	10.7923	11.3602	10.0305
$x_2$	2.0	2.0574	3.4391	1.9956	3.7000	3.6763	3.8298	3.1656
$y_2$	8.0	7.5175	7.7508	7.4669	8.1652	8.3575	8.8034	7.7928
$d_1$	0.5	0.5673	0.5976	0.5771	0.5897	0.5997	0.5974	0.5810
$d_2$	3.0	2.9874	4.4178	2.8647	4.7511	4.6970	4.7810	3.4644
$d_3$	2.0	2.2143	2.7132	1.9861	2.8805	2.6433	2.4767	1.7634
$p$	6.0	5.9078	3.7362	6.6373	3.6727	4.2654	4.7659	5.0759
$q$	-6.0	-6.1318	-7.6875	-5.3972	-8.2400	-8.3009	-8.6725	-6.9104
$E_r$	1.3696	0.3063	0.1621	0.2275	0.1633	0.1618	0.1617	0.1702

Table 2: Mechanism parameters and rms errors for problem 1

the tabugrad method, performs better. This together with the Powell, MATLAB and NAG methods perform equally well in terms  $E_r$ , with their values differing by less than 1%.. It is interesting to note however, that the mechanisms themselves show some variation; for example, the value of  $y_1$  varies between 10.00 and 11.36 over the results from these four methods.

It was noted that the seed mechanism used here was not the one from the catalogue that was deemed to give the closest match. The best one had the dyad formed by the coupler and driven links in the other configuration. This is shown on the left in figure 4. The figure also shows the results of using this as the seed for the search methods and the details are given in table 3 (although no comparison with other published results is possible). Again the

MATLAB and NAG produce the smallest rms errors and these are better than the results from the previous seed, with a reduction of 10% in  $E_r$ . However, this reduced error is at the cost of a larger mechanism.

	Seed	HJ	Powell	MATLAB	NAG	GANSO
$x_1$	0.3889	2.0703	2.0670	1.5553	1.5330	2.0929
$y_1$	5.8500	7.2754	7.1050	16.5036	16.9794	7.1140
$x_2$	2.6508	2.6873	1.9338	-1.9617	-2.0713	2.5652
$y_2$	5.0820	5.7306	5.6060	11.2502	11.5858	5.5654
$d_1$	0.7978	0.5338	0.4811	0.5937	0.5909	0.5389
$d_2$	2.3944	2.1669	2.3903	8.1221	8.2254	2.3055
$d_3$	2.3944	1.7901	1.5665	2.4013	2.3350	1.8650
$p$	4.7888	4.8048	3.8584	9.8547	10.2748	4.4323
$q$	1.1972	3.3426	4.1665	11.3303	11.5925	3.5912
$E$	0.9772	0.2351	0.2274	0.1446	0.1446	0.2299

Table 3: Mechanism parameters and rms errors for problem 1 with reversed dyad

## 4.2 Problem 2: simple path match (18 points)

The second problem involves a path with 18 points [11]. These are given in table 4 and shown in figure 5. Again angles are associated with each point, but now these only represent relative values between crank angles. The starting point for the angles can vary and there are now 10 degrees of freedom to work with.

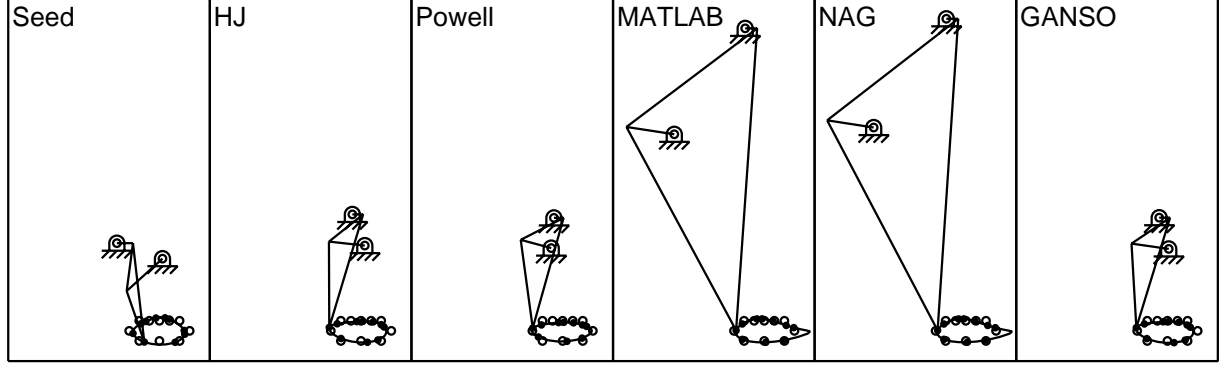


Figure 4: Mechanisms for problem 1 with reversed dyad

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$x$	0.5	0.4	0.3	0.2	0.1	0.05	0.02	0.0	0.0	0.03	0.1	0.15	0.2	0.3	0.4	0.5	0.6	0.6
$y$	1.1	1.1	1.1	1.0	0.9	0.75	0.6	0.5	0.4	0.3	0.25	0.2	0.3	0.4	0.5	0.7	0.9	1.0
$\theta$	30	50	70	90	110	130	150	170	190	210	230	250	270	290	310	330	350	370

Table 4: Precision points and prescribed crank angles for problem 2

The resultant mechanisms (together with the seed used for the search methods) are given in table 5 and are shown in figure 6.

Comparison is now with the results of using the tabugrad method [11]. It is clear that while this method has found a result with a low rms error, the mechanism obtained has a very long driven link, which may not be applicable in practice. The other methods also generate low rms error with Powell, MATLAB, NAG and GANSO all out-performing the tabugrad result. In fact, the last three of these converge to essentially the same mechanism. The value of  $E_r$  for MATLAB and NAG is 26% smaller than the value for tabugrad.



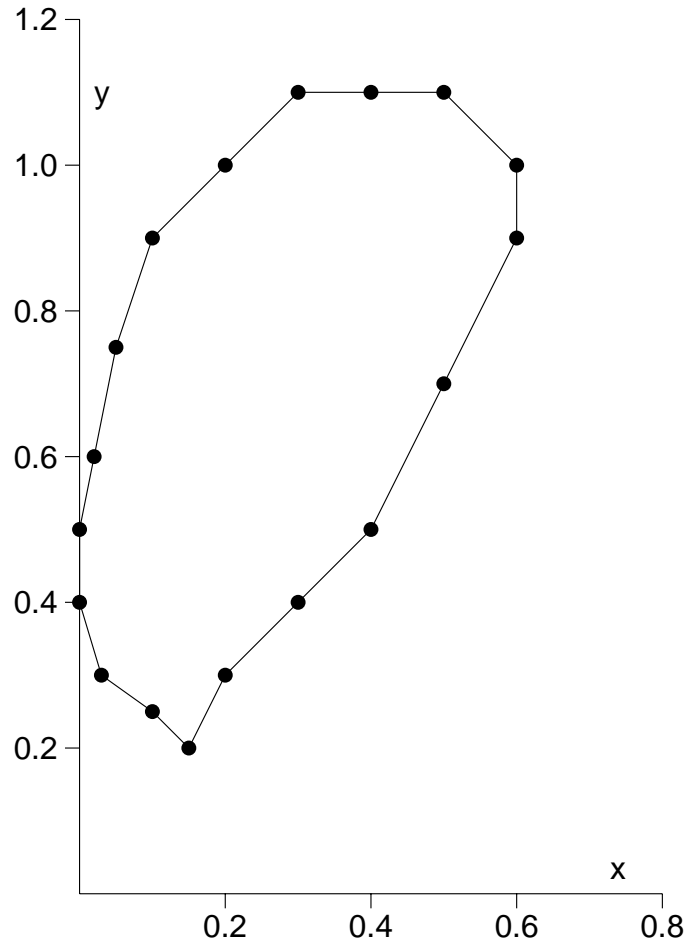


Figure 5: Precision points forming path for problem 2

### 4.3 Problem 3: transfer mechanism (25 points)

A trajectory within a packaging machine [41] provides the 25 precision points for this final case [10, 11, 13]. The points are given in table 6 and are plotted in figure 7. Crank positions are not specified and so the optimization process involves a total of 34 degrees of freedom.

Comparison is now made with results from the GA-FL method [10, 11] and (what is here referred to as) the SD method [13].

Table 7 gives the mechanism parameters for the various solution obtained, and table 8 provides the crank angles corresponding to each of the 25 precision points.

	Seed	Tabugrad	HJ	Powell	MATLAB	NAG	GANSO
$x_1$	0.7041	0.5856	0.7275	0.7474	0.5586	0.5586	0.5478
$y_1$	0.8712	0.7346	0.8004	0.7449	0.7188	0.7188	0.7214
$x_2$	1.5560	-9.5300	1.6446	1.6273	1.3193	1.3193	1.5617
$y_2$	0.1040	-0.9981	0.2054	0.2554	0.4403	0.4403	0.4843
$d_1$	0.3818	0.3297	0.3427	0.3176	0.3132	0.3131	0.3174
$d_2$	0.7636	0.5012	0.7002	0.6660	0.4445	0.4445	0.4377
$d_3$	1.1453	10.1910	1.0284	0.7222	0.7147	0.7147	0.9921
$p$	0.3818	0.1000	0.2038	-0.0954	-0.0264	-0.0264	0.0087
$q$	-0.3818	-0.3805	-0.4942	-0.5306	-0.3670	-0.3670	-0.3592
$\alpha$	-0.2765	0.2363	-0.3798	0.4609	0.3340	0.3348	0.2075
$E_r$	0.0446	0.0308	0.0332	0.0270	0.0226	0.0226	0.0232

Table 5: Mechanism parameters and rms errors for problem 2

All methods succeed in creating small rms error values. However all the “traditional” methods succeed in finding smaller errors than two adaptive approaches. In particular, the MATLAB and NAG find the best results, with reductions in  $E_r$  of 59% and 68% compared to GA-FL, although interestingly the mechanisms are significantly different.

## 5 Discussion and conclusions

The mechanism synthesis problem is critical for the design of high performance machinery. It can be formulated mathematically as an optimization problem in which the free variables

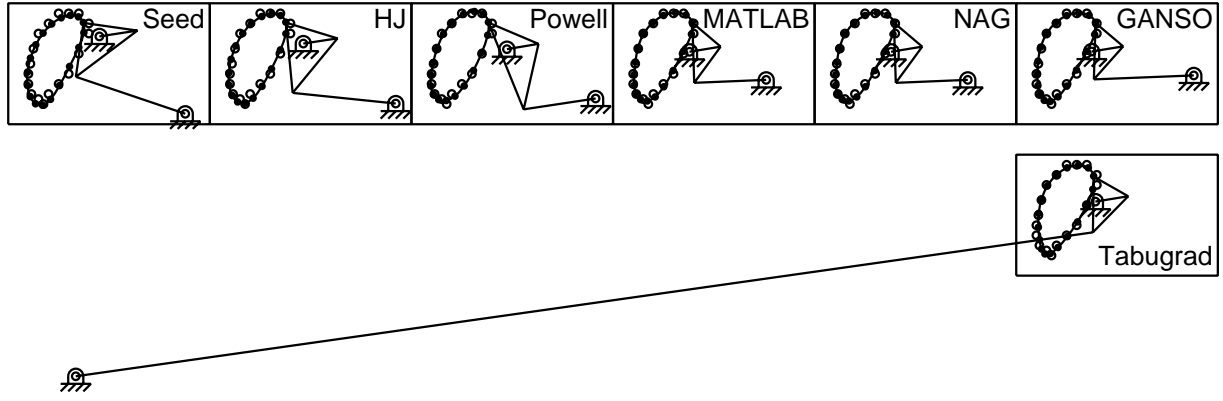


Figure 6: Mechanisms for problem 2

(7.03,5.99)	(5.43,3.56)	(3.76,1.22)	(3.76,4.91)	(5.07,6.85)
(6.95,5.45)	(4.93,2.94)	(3.76,1.97)	(3.76,5.47)	(5.45,6.84)
(6.77,5.03)	(4.67,2.60)	(3.76,2.78)	(3.80,5.98)	(5.89,6.83)
(6.40,4.60)	(4.38,2.20)	(3.76,3.56)	(4.07,6.40)	(6.41,6.80)
(5.91,4.03)	(4.04,1.67)	(3.76,4.34)	(4.53,6.75)	(6.92,6.58)

Table 6: Precision points for problem 3

are the parameters of the mechanism itself and, possibly, the position of the drive element when each precision point is attained. A number of numerical optimization schemes are available for attempting to handle the problem. These range from the “traditional” direct search and gradient through to the more recent adaptive methods such as genetic algorithms and simulated annealing.

A range of these methods has been applied to three case study examples. The aim has been to investigate whether the adaptive approaches offer advantages over the “traditional” ones. What has emerged is that there is no such advantage, at least in the cases consid-

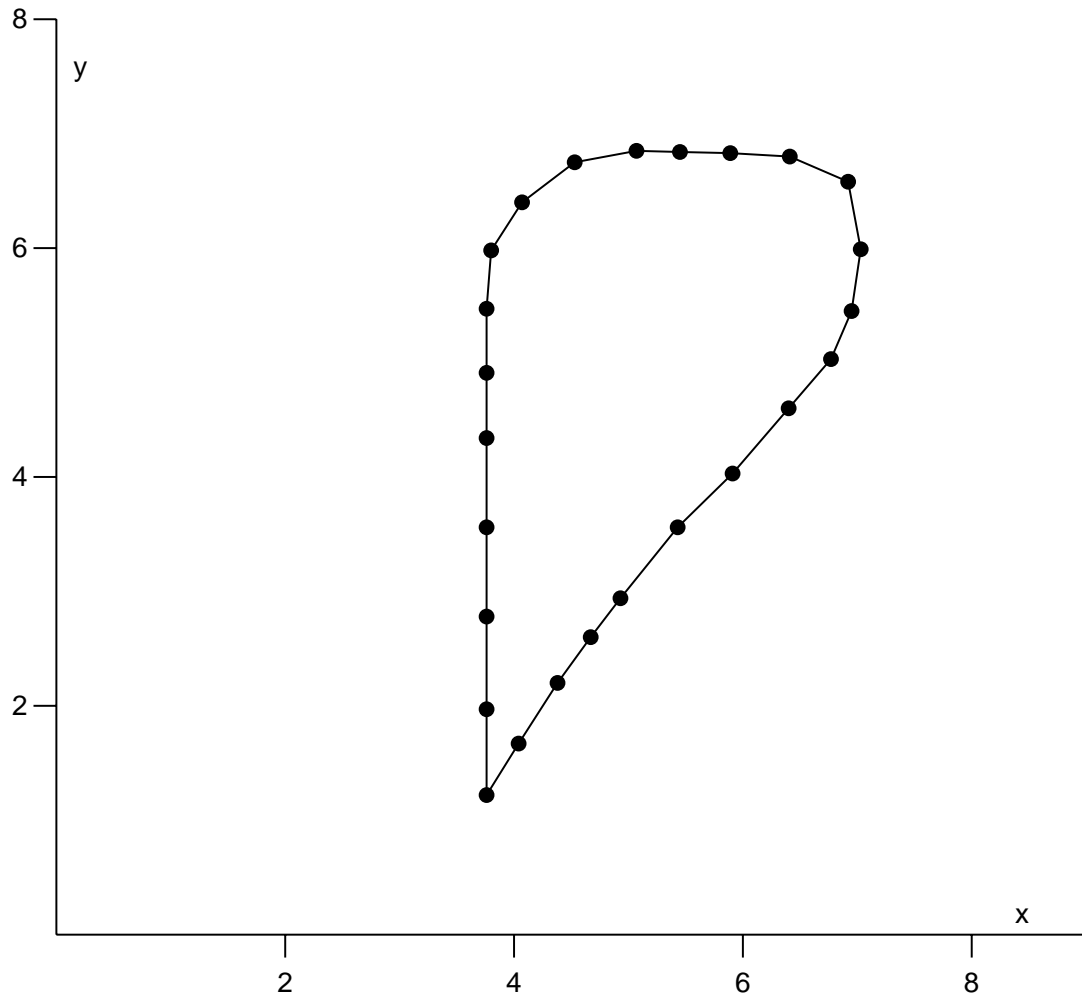


Figure 7: Precision points forming path for problem 3

ered. All methods succeeded in finding mechanisms with low rms values. However, some of the direct search and gradient methods, notably those used from the MATLAB and NAG packages, always performed better.

It may be argued that adaptive methods are less good at converging to a particular solution but are better at covering the design space and hence locating a global optimum (rather than a local one). They are thus less dependent on the choice of a good seed mechanism as a starting point. However, in the case of the first case study, all methods

	Seed	GA-FL	SD	HJ	Powell	MATLAB	NAG	GANSO
$x_1$	-4.1335	-2.4000	-5.5956	-4.2735	-3.8363	-2.2700	-1.7706	-3.8361
$y_1$	-3.5471	-4.0000	-1.1518	-3.2271	-0.8420	-1.2899	1.6308	-2.9365
$x_2$	5.8922	5.5470	3.9317	6.0822	4.2159	5.7418	5.7268	4.4178
$y_2$	-1.3847	0.2250	0.6459	-1.0347	0.6799	0.4321	0.1493	0.4288
$d_1$	2.5601	3.0100	2.5086	2.7201	2.5195	2.7214	2.3465	2.6640
$d_2$	7.7000	8.8000	7.8675	7.9300	6.3551	6.2997	4.3420	7.7646
$d_3$	7.7000	8.8000	5.0408	7.7400	5.2433	7.3845	5.6466	6.6313
$p$	11.5500	8.6260	11.8209	11.5400	9.9059	7.7251	7.4732	10.2653
$q$	-3.8500	-6.9850	-2.5633	-3.7900	-3.2216	-5.0724	-1.4000	-5.3416
$E_r$	0.1935	0.1584	0.1878	0.1331	0.0736	0.0646	0.0511	0.1010

Table 7: Mechanism parameters and rms errors for problem 3

succeeded in finding low rms mechanisms when starting from a particular seed. But the adaptive methods failed to find a better alternative which was found using the “traditional” methods when starting from a different seed (obtainable via a catalogue of mechanisms).

It is concluded therefore that the “traditional” direct search and gradient methods act as a good general-purpose tool. They are straightforward to implement and succeed in finding good solutions. In practice, a lot of mechanism design is variational, with the designer trying to adapt an existing design to a new application. In such cases, the current design can act as a suitable seed from which to start. The methods tend to stay close to this seed and thus are likely to provide a solution with which the designer is familiar and confident.

In cases where a range search is required a combination of approaches seems natural. An

	Seed	GA-FL	SD	HJ	Powell	MATLAB	NAG	GANSO
1	58.50	63.41	39.21	61.99	43.07	53.98	36.55	57.82
2	73.13	80.61	59.23	76.35	60.19	68.15	53.32	72.80
3	84.05	91.31	71.11	86.39	71.90	78.49	63.98	83.63
4	96.23	102.78	84.15	97.72	84.48	89.98	75.35	95.77
5	111.34	116.69	99.68	111.80	99.40	104.03	88.58	110.78
6	124.32	128.79	112.88	123.95	111.97	116.19	99.30	123.85
7	140.41	143.59	128.40	138.92	126.77	131.00	111.24	140.01
8	149.88	152.15	136.98	147.58	134.97	139.49	117.42	149.50
9	162.59	163.02	147.43	158.82	144.96	150.23	124.40	162.07
10	190.00	180.00	162.85	223.40	224.36	167.09	133.12	188.95
11	216.77	205.09	184.94	206.40	208.55	210.52	206.29	214.16
12	233.28	265.12	225.87	236.37	232.09	245.21	219.62	235.07
13	256.20	285.07	244.18	257.85	249.91	266.06	234.58	259.62
14	274.76	301.36	260.00	275.45	265.36	282.98	249.53	277.91
15	293.06	316.76	275.62	292.57	280.62	299.06	265.32	294.78
16	307.00	327.90	287.22	305.22	292.25	310.98	277.88	307.18
17	320.32	338.40	298.45	317.21	304.21	322.87	291.44	319.38
18	330.87	347.10	308.13	327.07	315.17	333.19	304.18	329.86
19	340.96	355.44	317.75	336.67	326.03	343.04	316.38	340.04
20	352.07	4.72	328.76	347.45	338.36	353.86	329.26	351.48
21	2.38	13.21	339.12	357.43	349.52	3.63	340.29	2.06
22	9.78	19.29	346.64	4.68	357.18	10.50	347.70	9.58
23	19.04	26.91	355.96	14.08	6.09	18.80	356.44	18.72
24	30.54	36.37	7.32	26.60	16.34	28.72	6.97	29.67
25	43.34	47.61	20.63	42.38	27.95	39.88	19.45	42.13

Table 8: Crank angular positions (in degrees) for problem 3

adaptive method can be used to find the general region in which a good (global) optimum lies, and then a “traditional” method employed to find this more precisely.

## Acknowledgements

The work reported in this paper was undertaken as part of the research activities of the Innovative Design and Research Centre at the University of Bath. This is supported by the Engineering and Physical Sciences Research Council (EPSRC) and this support is gratefully

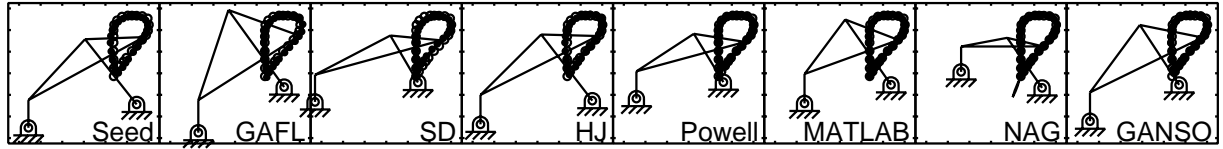


Figure 8: Mechanisms for problem 3

acknowledged.

## References

- [1] Nokleby, S. B. and Podhorodeski, R. P., 2001, “Optimization-based synthesis of Grashof’s geared five-bar mechanisms,” *Journal of Mechanical Design*, **123**, pp. 529–534.
- [2] Uicker, J. J., Pennock, G. R. and Shigley, J. E., 2003, *Theory of Mechanisms and Machines*, 3rd edition, Oxford University Press, New York.
- [3] Al-Smadi, Y. M., Russell, K. and Sodhi, R. S., 2009, “Planar four-bar motion generation with static structural conditions,” *Journal of Mechanisms and Robotics*, **1**, pp. 031009:1–6.
- [4] Al-Smadi, Y. M., Shen, Q., Russell, K. and Sodhi, R. S., 2009, “Planar four-bar motion generation with prescribed static torque and rigid-body reaction force,” *Mechanics Based Design of Structures and Machines*, **37**, pp. 73–85.
- [5] Angeles, J., Alivizatos, A. and Akhras, A., 1988, “Unconstrained non-linear least squares method of optimization of four-bar planar path generators,” *Mechanism and*

Machine Theory. **23**, pp. 343–353.

- [6] Cossalter, V., Doria, A., Pasini, M. and Scattlo, C., 1992, “Simple numerical approach for optimum synthesis of a class of planar mechanisms,” *Mechanism and Machine Theory*, **27**, pp. 357–366.
- [7] Mariappan, J. and Krishnamurty, S., 1996, “A generalized exact gradient method for mechanism synthesis,” *Mechanism and Machine Theory*, **31**, pp. 413–421.
- [8] Sancibrian, R., Viadero, P., Garcíá, P. and Fernández, A., 2004, “Gradient-based optimization of path synthesis problems in planar mechanisms,” *Mechanism and Machine Theory*, **39**, pp. 839–856.
- [9] Cabrera, J. A., Simon, A. and Prado, M., 2002, “Optimal synthesis of mechanisms with genetic algorithms,” *Mechanism and Machine Theory*, **37**, pp. 1165–1177.
- [10] Laribi, M. A., Mlika, A., Romdhane, L., and Zeghloul, S., 2004, “A combined genetic algorithm-fuzzy logic method (GA-FL) in mechanism synthesis,” *Mechanism and Machine Theory*, **39**, pp. 717–735.
- [11] Smaili, A., Diab, N. and Atallah, N., 2005, “Optimal synthesis of mechanisms using tabu-gradient search algorithm,” *Journal of Mechanical Design*, **127**, pp. 917–923.
- [12] Liu, Y. and Xiao, R., 2005, “Optimal synthesis of mechanisms for path generation using refined numerical representation based model and AIS based searching method,” *Journal of Mechanical Design*, **127**, pp. 688–691.



- [13] Smaili, A. and Diab, N., 2007, “A new approach to shape optimization for closed path synthesis of planar mechanisms,” *Journal of Mechanical Design*, **129**, pp. 941–948.
- [14] Kinzel, E. C., Schmiedeler, J. P. and Pennock, G. R., 2006, “Kinematic synthesis for finitely separated positions using geometric constraint programming,” *Journal of Mechanical Design*, **128**, pp. 1070–1079.
- [15] Podgorelec, D., 2002, “A new constructive approach to constraint-based geometric design,” *Computer-Aided Design*, **34**, pp. 769–785.
- [16] Lin, L. and Chen, L. C., 2002, “Constraints modelling in product design,” *Journal of Engineering Design*, **13**, pp. 205–214.
- [17] McMahon, C., Lowe, A., Culley, S., Corderoy, M., Crossland, R., Shah, T. and Stewart, D., 2004, “Waypoint: an integrated search and retrieval system for engineering documents,” *Journal of Computing and Information Science in Engineering*, **4**, pp. 329–338.
- [18] Hicks, B. J., Medland, A. J. and Mullineux, G., 2006, “The representation and handling of constraints for the design, analysis, and optimization of high speed machinery,” *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **20** pp. 313–328.
- [19] Lallouet, A. and Legtchenko, A., 2006, “Partially defined constraints in constraint-based design,” *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **20**, pp. 297–311.
- [20] Awtar, S. and Slocum, A. H., 2007, “Constraint-based design of parallel kinematic XY flexure mechanisms,” *Journal of Mechanical Design*, **129** pp. 816–830.

- [21] Ge, J.-X., Chou, S.-C. and Gao, X.-S., 1999, "Geometric constraint satisfaction using optimization methods," *Computer-Aided Design*, **31**, pp. 867–879.
- [22] Mullineux, G., 2001, "Constraint resolution using optimization techniques," *Computers and Graphics*, **25**, pp. 483–492.
- [23] Martínez, M. L. and Félez, J., 2005, "A constraint solver to define correctly dimensioned and over-dimensioned parts," *Computer-Aided Design*, **37**, pp. 1353–1369.
- [24] Hoffmann, C. M., 2005, "Constraint-based computer aided design," *Journal of Computing and Information Science in Engineering*, **5**, pp. 182–187.
- [25] Sitharam, M., Oung, J.-J., Zhou, Y. and Arbree, A., 2006, "Geometric constraints within feature hierarchies," *Computer-Aided Design*, **38**, pp. 22–38.
- [26] Ma, W., Zhong, Y., Tso, S.-K. and Zhou, T., 2004, "A hierarchically structured and constraint-based data model for intuitive and precise solid modelling in a virtual reality environment," *Computer-Aided Design*, **36**, pp. 903–928.
- [27] Mullineux, G., Hicks, B. J. and Medland A. J., 2005, "Constraint-aided product design," *Acta Polytechnia*, **45**, pp. 31–36.
- [28] Zhang, Y. M., He, X. D., Liu, Q. L. and Wen, B. C., 2005, "An approach of robust reliability design for mechanical components," *Proc. Instn. Mech. Engrs, Part E: Journal of Process Mechanical Engineering*, **219** pp. 275–283.
- [29] Walsh, G., 1975, *Methods of Optimization*, Wiley, London.
- [30] Fletcher, R., 1987, *Practical methods of Optimization*, 2nd ed., John Wiley, Chichester.

- [31] Papalambros, P. and Wilde, D., 2000, *Principles of Optimal Design*, 2nd ed., Cambridge University Press, Cambridge.
- [32] Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading.
- [33] Parmee, I. C., ed., 2002, *Adaptive Computing in Design and Manufacture V*, Springer, London.
- [34] Eiben, A. E. and Smith, J. E., 2003, *Introduction to Evolutionary Computing*, Springer, Berlin.
- [35] Ashlock, D., 2006. *Evolutionary Computation for Modelling and Optimization*, Springer, Berlin.
- [36] MathsWorks, 2010, “Optimization toolbox”, <http://www.mathworks.com/products/optimization/> (accessed January 2010).
- [37] Numerical Algorithms Group, 2010, “NAG C Library”, <http://www.nag.co.uk> (accessed January 2010).
- [38] GANSO, 2010, “Programming library for global and non-smooth optimization” University of Ballarat, <http://www.ganso.com.au/> (accessed January 2010).
- [39] Bagirov, A. M., Ghosh, M. and Webb, D., 2006, “A derivative-free method for linearly constrained, non-smooth optimization, *Industrial and Management Optimization*, **2**, pp. 319–338.

- [40] Singh, B., Matthews, J., Mullineux, G. and Hicks, B. J., 2008, “Design catalogues for mechanism selection”, in: *Proc. DESIGN 2008, 10th International Design Conference*, Marjanović, D., Štorga, N., Pavković, N. and Bojčetić, N. eds., Dubrovnik, pp. 673–680.
- [41] McGarva, J. R., 1994, “Rapid search and selection of path generating mechanisms from a library,” *Mechanism and Machine Theory*, **29** pp. 223–235.

## List of Figure Captions

1. Figure 1: Four bar mechanism and notation
2. Figure 2: Precision points forming path for problem 1
3. Figure 3: Mechanisms for problem 1
4. Figure 4: Mechanisms for problem 1 with reversed dyad
5. Figure 5: Precision points forming path for problem 2
6. Figure 6: Mechanisms for problem 2
7. Figure 7: Precision points forming path for problem 3
8. Figure 8: Mechanisms for problem 3

## List of Table Captions

1. Table 1: Precision points and prescribed crank angles for problem 1
2. Table 2: Mechanism parameters and rms errors for problem 1
3. Table 3: Mechanism parameters and rms errors for problem 1 with reversed dyad
4. Table 4: Precision points and prescribed crank angles for problem 2
5. Table 5: Mechanism parameters and rms errors for problem 2
6. Table 6: Precision points for problem 3
7. Table 7: Mechanism parameters and rms errors for problem 3
8. Table 8: Crank angular positions (in degrees) for problem 3